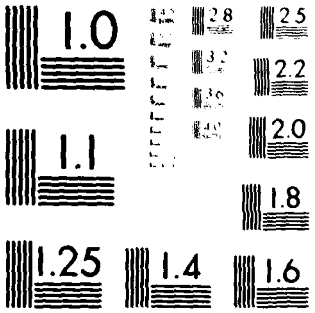
AD-A082 735 SYSTEM DEVELOPMENT CORP HUNTSVILLE ALA F/G 9/2
REDIMEN: SYSTEMS ANALYSIS OF INTEGRATED NETWORKS OF TASKS (SAIN--ETC(U)
FEB 80 D J SEIFERT, G KOEPLINGER, C M HOYLAND F33615-79-C-0505
UNCLASSIFIED SDC-TM-HU-539/000/00 AMRL-TR-80-5 NL

1 of 1
AD
A082735

END
DATE
FILMED
5-80
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963-A

Systems Analysis of Integrated Networks of Tasks

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| | | |
|---|---|---|
| (18) REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS<br>BEFORE COMPLETING FORM |
| 1. REPORT NUMBER (19)<br>AFAMRL-TR-80-5 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>(6) REDIMEN: SAINT REDIMENSIONING PROGRAM. | | 5. TYPE OF REPORT & PERIOD COVERED<br>(9) Technical Report<br>July 1977—October 1979 |
| | | 6. PERFORMING ORG. REPORT NUMBER<br>(14) SDC-TM-HU-539/000/00 |
| 7. AUTHOR(s)<br>(10) Deborah J. Seifert<br>George Koeplinger*<br>Constance M. Hovland** | | 8. CONTRACT OR GRANT NUMBER(s)<br>(15) F33615-79-C-0505 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Air Force Aerospace Medical Research Laboratory<br>Aerospace Medical Division, AFSC<br>Wright-Patterson AFB OH 45433 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>(16) (17)<br>62202F; 7184-10-24 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS | | 12. REPORT DATE<br>(11) February 1980 |
| | | 13. NUMBER OF PAGES<br>20 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)<br>(12) 20 | | 15. SECURITY CLASS. (of this report)<br>Unclassified |
| | | 15a. DECLASSIFICATION DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DTIC ELECTE APR 2 1980 S D B

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

* UDRI
University of Dayton
Dayton OH 45409

** SDC/Suite 305
4134 Linden Avenue
Dayton OH 45432

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Simulation, Networks, Discrete Event Simulation, Next Event Simulation, Continuous Simulation, Combined Simulation, Simulation Languages, Modeling, Operations Research, Mission Analysis, Man Machine Systems, Crew Performance, SAINT

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The SAINT (Systems Analysis of Integrated Networks of Tasks) computer programs have fixed storage array dimensions. Different models may have storage requirements less than (or greater than) the presently defined array sizes in SAINT, which makes the SAINT programs inefficient (or impossible to run) unless the common statements in all of the subroutines are redimensioned. This is a laborious, error-prone task.

(Continued)

405247

20. Abstract

REDIMEN was designed and implemented to eliminate most of the manual labor required to revise the SAINT source codes to tailor array sizes to the model or problem at hand. This document describes the REDIMEN program, its source code, necessary inputs, and its utilization in conjunction with SAINT applications efforts.

## PREFACE

This report describes the FORTRAN program REDIMEN (REDIMENSIONING PROGRAM). REDIMEN was written to facilitate the redimensioning of the common variables in SAINT (Systems Analysis of Integrated Network of Tasks) to correspond to individual modeling requirements.

The initial concepts and detailed design of REDIMEN were formulated by Deborah J. Seifert. Preliminary implementation was accomplished by George Koeplinger. Initial use and minor modifications to the programs were accomplished by Constance M. Hoyland who also prepared all of the program documentation.

## INTRODUCTION

The SAINT program that is available for distribution has default array sizes to accommodate a model with certain characteristics. However, the model characteristics can be altered by changing the values of variables in the main program representing these default values and the dimensions of the associated arrays in the common blocks and all SAINT subprograms where their associated COMMON blocks appear.

REDIMEN was written to facilitate these changes. The user has only to provide the maximum value for each variable and the source program for SAINT as input. REDIMEN creates a new SAINT source program with the user-specified parameters and dimensions as output. Figure 1 shows the logic flow for operating the REDIMEN program.

## INPUTS

Table 1 contains the SAINT maximum value variables, their definition, and their default values. (Definitions of all SAINT COMMON variables can be found in Table 2.)

The model characteristics that the SAINT program will accept can be altered by changing these variables. For example, assume that a SAINT model requires that the largest task number be 500. Entry 3 of Table 1 indicates that the name of the variable that is set to the largest task number is IMN (which is presently set to 100). Thus IMN must be changed to 500.

Values for the 34 maximum value variables are read into REDIMEN, Unit 5, one value to a card in an (I4) format. Be careful to maintain the exact format and sequence of these cards.

After REDIMEN reads in the maximum values, it begins reading the SAINT source, one 80 column card image at a time from Unit 8. REDIMEN determines whether a change in dimension size is required, and if so, makes the alteration before copying the record to the output file on Unit 9. This process continues until the entire SAINT source has been appropriately revised and copied.

## OUTPUT

The output file on Unit 9 contains a new SAINT source with the user-specified COMMON block dimensions. However, since REDIMEN only alters the array sizes, the definition

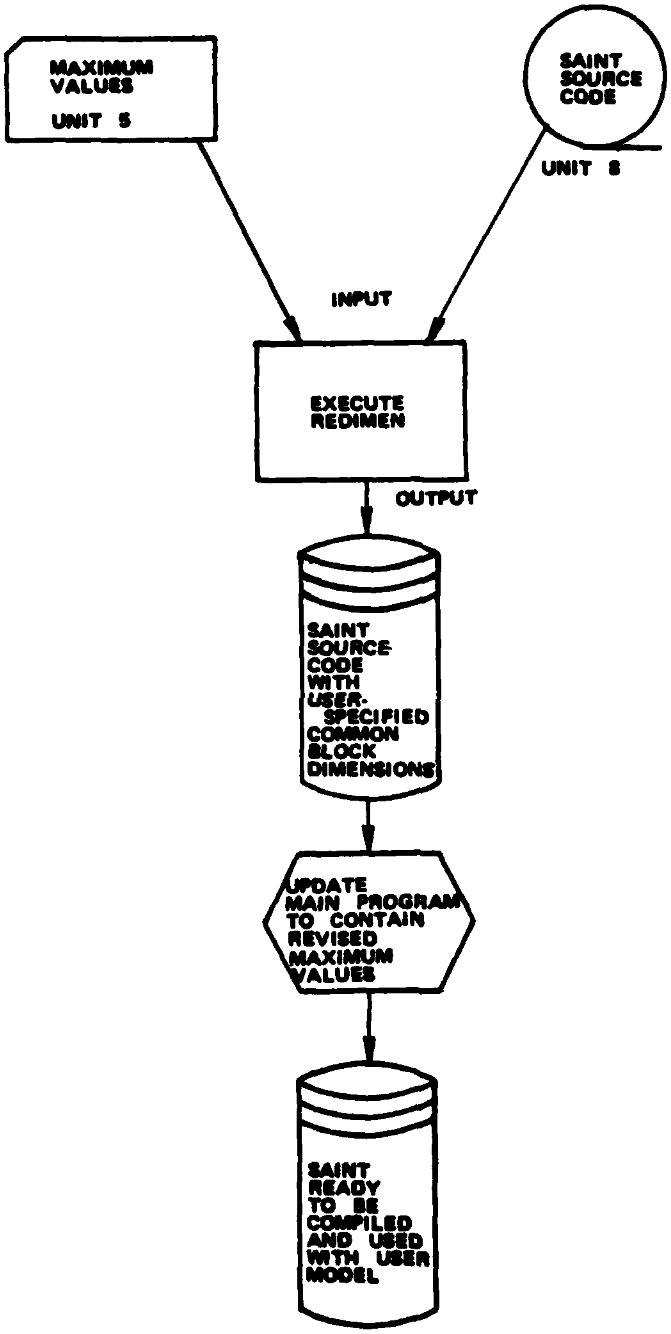

MAXIMUM VALUES
UNIT 5
SAINT SOURCE CODE
UNIT 8
INPUT
EXECUTE REDIMEN
OUTPUT
SAINT SOURCE CODE WITH USER-SPECIFIED COMMON BLOCK DIMENSIONS
UPDATE MAIN PROGRAM TO CONTAIN REVISED MAXIMUM VALUES
SAINT READY TO BE COMPILED AND USED WITH USER MODEL

Figure 1. Implementation for REDIMEN

## Table 1. Dimension Requirements for SAINT Arrays

| Controlling Characteristic | Current Maximum Value | Variable Name for Maximum Value | Labeled COMMON Block Containing Arrays Affected | Arrays Affected |
|---|---|---|---|---|
| 1. Number of tasks being performed + number of tasks awaiting assignment of resources. | 100 | ID | COM02 | NSET(ID*4), QSET(ID*3) |
| 2. Largest distribution set number. | 100 | MPARM | COM09 | All dimensioned arrays in COM09 |
| 3. Largest task number. | 100 | IMN | COM10 | All dimensioned arrays in COM10 |
| 4. Largest resource number. | 20 | MOPNO | COM11 | BUSY, LLRES,NBUS,NOPTR,TLST,<br>RSTAT(MOPNO*4) |
| 5. Number of tasks that require resources + total number of task-resource associations specified. | 600 | MNOPA | COM11 | NOPA |
| 6. (Number of tasks that have branches emanating from their output side * 2) + number of tasks which require probabilistic branching + total number of deterministic branches specified + (total number of probabilistic branches specified * 2) + (total number of conditional branches specified * 5). | 850 | MVARA | COM12 | VARA |
| 7. Number of tasks that cause task modifications + (total number of task modifications specified * 2) + number of tasks that cause distribution set modifications + (total number of distribution set modifications specified * 2) + number of tasks that cause task clearing + (total number of task clearings specified * 2) + number of tasks that cause resource clearing + (total number of resource clearings * 2) + number of tasks that require different predecessors + total number of predecessors to tasks that require different predecessors. | 250 | IMNA | COM12 | NABA |

Table 1 (Continued)

| Controlling Characteristic | Current Maximum Value | Variable Name for Maximum Value | Labeled COMMON Block Containing Arrays Affected | Arrays Affected |
|---|---|---|---|---|
| 8. Number of user-defined task characteristics for all tasks (the number of user-defined task characteristics for a task is equal to the largest characteristic number defined for that task). | 200 | MTCHR | COM12 | STCHR |
| 9. Largest moderator function number. | 20 | MMDFN | COM13 | MDFNS,MFSTM |
| 10. Number of tasks at which moderator function status is updated + (total number of moderator function status updates specified * 2). | 300 | MMSTU | COM13 | MFSTU |
| 11. Number of statistics tasks. | 50 | MXSTA | COM14 | MSINK,KSTPE,KSTTM,XSTUS,NCELS,<br>XLOW, WIDTH, SUMAI, SUMAF |
| 12. (Number of statistics tasks * 2) + total number of cells specified for statistics task histograms. | 1350 | MNCLS | COM14 | JCELS |
| 13. Number of information packets in the network * (number of attributes per packet + 1). | 1000 | MAXU6 | COM15 | DESCR |
| 14. Largest resource number * number of resource attributes per resource. | 100 | MDOAT | COM15 | DOATT |
| 15. (Number of tasks at which attribute assignments are to be made * 2) + (total number of attribute assignments to be made * 4). | 800 | MDSTR | COM15 | NDSTR |
| 16. Largest system attribute number. | 100 | MSYAT | COM15 | SYSAT |
| 17. Largest index for state variable equations (SS(·) or DD(·) variables). | 100 | MEQT | COM17 | All dimensioned arrays in COM17[1] |
| 18. Largest switch number. | 20 | MNSWA | COM18 | IS |

Table 1 (Continued)

| Controlling Characteristic | Current Maximum Value | Variable Name for Maximum Value | Labeled COMMON Block Containing Arrays Affected | Arrays Affected |
|---|---|---|---|---|
| 19. Number of tasks at which switch values are changed + (total number of specified switch value changes as a result of task completion * 2). | 300 | MDAD | COM18 | NABAD |
| 20. Number of tasks at which state variable values are regulated + (total number of specified state variable regulations as a result of task completion * 5). | 600 | MDDR | COM18 | YABAR |
| 21. Largest state variable monitor number. | 20 | MPFLAG | COM19 | IPFLAG,NPOSS,NPOST,LLMON,<br>THRES(MPFLAG*6) |
| 22. Number of state variable monitors causing tasks to be signaled + total number of specified task signalings as a result of monitor action. | 40 | MDNPT | COM19 | NABAT |
| 23. Number of state variable monitors causing switch values to be changed + (total number of specified switch value changes as a result of monitor action * 2). | 60 | MDNSS | COM19 | NABAS |
| 24. Number of state variables for which statistics are to be collected. | 20 | MSTAT | COM20 | All dimensioned arrays in COM20 |
| 25. Largest state variable plot number. | 10 | MPLOT | COM21 | DTPLT,IITAP,NNPTS,NNVAR,NNVP,<br>NVP(MPLOT*MNVPP) |
| 26. Largest number of variables plotted on any state variable plot. | 10 | MNVPP | COM21 | LLPHI,LLPLO,LLSYM,PPHI,PPLO,<br>NVP(MPLOT*MNVPP),LLSVP(MNVPP + 1,2) |
| 27. Number of state variable core plot point sets * (number of variables being plotted + 1). | 1100 | MDQP | COM21 | QPSET |

Table 1 (Continued)

| Controlling Characteristic | Current Maximum Value | Variable Name for Maximum Value | Labeled COMMON Block Containing Arrays Affected | Arrays Affected |
|---|---|---|---|---|
| 28. Largest statistic number for user-generated statistics based on observation. | 20 | MNCLT | COM23 | LLUGC,USOBV |
| 29. Largest statistic number for user-generated statistics based on time-persistent variables. | 20 | MNSTP | COM23 | LLUGT,TTCLR,USTPV |
| 30. Largest user-generated histogram number. | 20 | MNHIS | COM23 | LLUGH,MNCEL,MNLOW,MNWID |
| 31. (Number of user-generated histograms * 2) + total number of cells specified for user-generated histograms. | 540 | MNCEL | COM23 | JJCEL |
| 32. Largest user-generated plot number. | 10 | MNPLT | COM24 | DPLOT,ITAPE,MPTSV,NVARS |
| 33. Largest number of variables plotted on any user-generated plot. | 10 | MNVAR | COM24 | LPMIN,LPLOW,LSYMB,PMIN,PLOW,<br>LLUGP(MNVAR+1,2) |
| 34. Number of user-generated core plot point sets * (number of variables being plotted + 1) | 1100 | MNCUP | COM24 | UPSET |

[1] The following arrays appear in a DIMENSION statement in subroutine UPDATE:
S(,DI,A1,A2,A3,A4,A5,A6,A7
The dimensions of these arrays must be changed in accordance with any changes made to the maximum number of state variables allowed.

of the maximum value variables that occurs in MAIN have not been updated. Since these values must exactly match the definitions of the variables that were used as input to the REDIMEN program, they should be changed by the user before compiling SAINT.

Table 2 contains a listing of the SAINT MAIN program. The MAIN program contains all of the SAINT COMMON blocks with the correct user-specified dimensions. Since user-written subroutines have access to any SAINT variable through the inclusion of the appropriate SAINT COMMON block, the user may copy any of the COMMONs from MAIN for this purpose.

Table 2. MAIN Program for SAINT

```
 COMMON /COM01/ ID,IM,IMM,IMNA,IMN,MAXDS,MDAD,MDDR,MDNPT,MDNSS,      00000030
*               MDDAT,MDCP,MDSTR,MEQT,MFLAG,MMPTS,MMDFN,MMSTU,        00000040
*               MNCEL,MNCLS,MNCLT,MNCUP,MNHIS,MNOPA,MNPLT,MNPTQ,      00000050
*               MNSTP,MNSWA,MNVAR,MNVPP,MOPNO,MPARM,MPLOT,MSTAT,      00000060
*               MSYAT,MTCHR,MXSTA,MYABA,NUQ                           00000070
 COMMON /COM02/ ATRIB(3),JTRIB(2),QSET(300),NSET(400),MFA,MXX,        00000080
*               MFE(3),MLE(3),NQ(3)                                   00000090
 COMMON /COM03/ IPOS,JPOS,KPOS,LPOS,MPOS,NPA,NAN,IERRW,IERRF,IFIN,    00000100
*               IIECH,INDXS,INDXT,INDX,JNDX,KNDX,IP,NUMFL,ICONT,      00000110
*               IISED,HIVAL,IBLNK,IZERO,LA,LB,LC,LD,LE,LF,LG,LH,LI,00000120
*               LJ,LK,LL,LM,LN,LO,LP,LQ,LR,LS,LT,LU,LV,LW,LX,LY,LZ 00000130
 COMMON /COM04/ IDFAL(4),KREAD(40),IFLAG(50),IRSUL(50),RESUL(50),    00000140
*               IABC(8,50),KARD(80),IDIG(9)                           00000150
 COMMON /COM05/ NPROJ,MON,NDAY,NYR,NAME(2),NRUN,NRUNS,NSKSR,          00000160
*               NSKST,LLCVD,NNEQD,NNEQS,NNEQT                         00000170
 COMMON /COM06/ TNOW,TTNEX,MFAD,SEED,ISEED,NCRDR,NPRNT,NPUNCH,        00000180
*               NRNIT,NRENT,MNDC,NDC,NDTN,NNTC                        00000190
 COMMON /COM07/ NDE,NOPAT,NSYAT,NDUP,NNM,NNPA,NMDFN,NN,NPRMS,IFLPR,00000200
*               JFLPR,KRNK,XINN,NFLAG,NNCLT,NNHIS,NNPLT,NNSTA,        00000210
*               NNSTP,NPLOT,NSTTS                                     00000220
 COMMON /COM08/ NEIP,NEIS,NSIP,NSIS,ITRACE,JTRACE,NRTSP,NRTEP,        00000230
*               KTRACE,MTRACE,IIRSR,IISSR,IGRAF,JGRAF,IJTRAC,         00000240
*               NSVVS,NSVVE,NTSUE,NTSUS,LTRACE,NRTSS,NRTES            00000250
 COMMON /COM09/ PARAM(100,5),NPTBU(100),PARM1(100),PARM4(100)         00000260
 COMMON /COM10/ CACIN(100,3),ITCHR(100),LLTSK(100,2),LSINK(100),      00000270
*               MACIN(100,4),MFEN(100),MFSTT(100),NFTBU(100),         00000280
*               MOP(100),MPO(100),NDCH(100,2),NDEL(100),NDPT(100), 00000290
*               NPAR(100),NPD(100),NPDDR(100),NPDDS(100),NPUP(100),00000300
*               NPUR(100),NPSGN(100),NREL(100),NRELP(100),            00000310
*               NREL2(100),NSIGN(100),NTC(100),NTCHR(100),            00000320
*               NTYPE(100),KMARK(100),XMARK(100)                      00000330
 COMMON /COM11/ BUSY(20),LLRES(20,2),NBUS(20),NUPTR(20),TLST(20),     00000340
*               NOPA(600),RSTAT(80)                                   00000350
 CCMMON /COM12/ YABA(850),NABA(250),STCHR(200)                        00000360
 COMMON /COM13/ MDFNS(20),MFSTW(20),MFSTU(300)                        00000370
 COMMON /COM14/ NSINK(50),KSTPE(50),KSTTM(50),XSTUS(50),NCELS(50), 00000380
*               XLOW(50),WIDTH(50),SUMAI(50,5),SUMAF(50,5),           00000390
*               JCELS(1350)                                           00000400
 COMMON /COM15/ DESCR(1000),DJATT(100),NDSTR(800),SYSAT(100)          00000410
 COMMON /COM16/ AAERR,DTMAX,DTMIN,DTSAV,IITES,LLERR,KRERR,TTLAS,      00000420
*               TTSAV,DTSUG,DTFUL,DTNOW,ISEES,RESLS,DTACC,LLSAV,      00000430
*               LSAVE                                                 00000440
 COMMON /COM17/ SS(100),SSL(100),DD(100),DDL(100),LLSVR(100,2)        00000450
 COMMON /COM18/ IS(20),NABAD(300),YABAR(600)                          00000460
 COMMON /COM19/ LFLAG(20),NPOSS(20),NPOST(20),LLMON(20,2),            00000470
*               NABAT(40),NABAS(60),THRES(120)                        00000480
 COMMON /COM20/ NSTAI(20),LLSVS(20,2),SSTPV(20,6)                     00000490
 COMMON /COM21/ DTPLT(10),IITAP(10),NNPTS(10),NNVAR(10),NNVP(10),     00000500
*               LLPLT,NNPT,LLPHI(10),LLPLO(10),LLSYM(10),PPHI(10), 00000510
*               PPLO(10),NVP(100),LLSVP(11,2),QPSET(1100)             00000520
 COMMON /COM22/ TTIME,PFIRB                                           00000530
 COMMON /COM23/ LLUGC(20,2),USOBV(20,5),LLUGT(20,2),TTCLR(20),        00000540
*               USTPV(20,6),LLUGH(20,2),NNCEL(20),HHLOW(20),          00000550
*               HHWID(20),JJCEL(540)                                  00000560
 COMMON /COM24/ DPLOT(10),ITAPE(10),NPTSV(10),NVARS(10),LPLOT,        00000570
*               NPTEX,LPHIH(10),LPLOW(10),LSYMB(10),PHIH(10),         00000580
*               PLOW(10),LLUGP(11,2),UPSET(1100)                      00000590
```

## Table 2 (Continued)

```
C
      NCRDR=5
      NPRNT=6
      NPUNCH=7
      NRNIT=8
      NRENT=9
C
C*****DEFINE VARIABLES WHICH REPRESENT ARRAY MAXIMA AND SIMULATION
C*****LIMITS
C
      ID=100
      IMN=100
      IMNA=250
      MAXDS=1000
      MDAD=300
      MDDR=600
      MDNPT=40
      MDNSS=60
      MDOAT=100
      MDQP=1100
      MDSTR=800
      MEQT=100
      MFLAG=20
      MMDFN=20
      MMSTU=300
      MNCEL=540
      MNCLS=1350
      MNCLT=20
      MNCUP=1100
      MNHIS=20
      MNOPA=600
      MNPLT=10
      MNSTP=20
      MNSWA=20
      MNVAR=10
      MNVPP=10
      MOPNO=20
      MPARM=100
      MPLOT=10
      MSTAT=20
      MSYAT=100
      MTCHR=200
      MXSTA=50
      MYABA=850
C
C*****EXECUTION CYCLE -- READ INPUT DATA, THEN INITIATE THE SIMULATION
C
      IFIN=0
  100 CALL DATIN
      IF(IERRF.EQ.0) CALL GASP
      IF(IERRF.EQ.0) CALL SUMRY
      IF (IFIN.EQ.0) GO TO 100
      STOP
C
      END
```

# APPENDIX A - SOURCE CODE

Following is a listing of the REDIMEN source code including an example of the JCL necessary to implement it on the IBM 370 (see Table 3). Different installations would, of course, use control cards appropriate to the specific machine and operating system in use at that facility.

Following the listing are sample input data by which a SAINT model could be scaled down from the default values. The model defined would be discrete (no use of state variables) with 20 or less tasks. Redimensioning in this way would require less space when SAINT is executed.

## Table 3. Listing with JCL

```
//REDIMEN  JOB
//  EXEC  FORTGCLG
//FORT.SYSIN  DD  *
C                                 REDIMEN
C
C     PROGRAM REDIMEN SERVES TO REDIMENSION THE ARRAYS OF THE SAINT
C     COMMON BLOCKS THAT ARE AFFECTED BY THE MAXIMUM VALUE VARIABLES
C     (SEE "THE SAINT USER'S MANUAL" SECTION 10).
C
C     **IMPORTANT** EACH INPUT CARD CONTAINS EXACTLY "1" MAX VALUE
C     OF FORMAT(I4).
C
C
      INTEGER CMBL(15)/'M02/','M09/','M10/','M11/','M12/','M13/','M14/',
     1'M15/','M17/','M18/','M19/','M20/','M21/','M23/','M24/'/
      INTEGER TAPIN(18),M(34)
C
C
      DO 100 I=1,34
100   READ (5,350) M(I)
C
      M3=M(1)*3
      M4=M(1)*4
      M5=M(25)+1
      M21=M(27)*4
      M69=M(26)*M(29)
      M61=M(26)+1
      M70=M(13)*6
C
      DO 330 J=1,8500
      READ (8,360,END=340) TAPIN,INDX
      DO 110 I=1,15
      IF (TAPIN(5).EQ.CMBL(I)) GO TO 120
110   CONTINUE
      I=16
120   GO TO (130,150,160,180,190,200,210,240,250,260,270,280,290,300,310
     1,320), I
130   WRITE (9,370) M3,M4,INDX
      INDX=INDX+10
      WRITE (9,380) INDX
140   READ (8,360) TAPIN,INDX
      GO TO 330
150   WRITE (9,390) M(28),M(28),M(28),M(28),INDX
      GO TO 330
160   WRITE (9,400) M(2),M(2),M(2),M(2),INDX
      INDX=INDX+10
      WRITE (9,410) M(2),M(2),M(2),M(2),INDX
      INDX=INDX+10
      WRITE (9,420) M(2),M(2),M(2),M(2),M(2),INDX
      INDX=INDX+10
      WRITE (9,430) M(2),M(2),M(2),M(2),M(2),INDX
      INDX=INDX+10
      WRITE (9,440) M(2),M(2),M(2),M(2),INDX
      INDX=INDX+10
      WRITE (9,450) M(2),M(2),M(2),M(2),INDX
      INDX=INDX+10
      WRITE (9,460) M(2),M(2),M(2),INDX
      DO 170 K=1,6
170   READ (8,360) TAPIN,INDX
```

### Table 3 (Continued)

```
      GO TO 330
180   WRITE (9,470) M(27),M(27),M(27),M(27),M(27),INDX
      INDX=INDX+10
      WRITE (9,480) M(21),M21,INDX
      GO TO 140
190   WRITE (9,490) M(34),M(3),M(32),INDX
      GO TO 330
200   WRITE (9,500) M(14),M(14),M(15),INDX
      GO TO 330
210   WRITE(9,510) M(33),M(33),M(33),M(33),INDX
      INDX=INDX+10
      WRITE(9,520) M(33),M(33),M(33),M(33),INDX
      INDX=INDX+10
      WRITE(9,530) M(33),M(17),INDX
220   DO 230 K=1,2
230   READ (8,360) TAPIN,INDX
      GO TO 330
240   WRITE (9,540) M(4),M(9),M(11),M(31),INDX
      GO TO 330
250   WRITE (9,550) M(12),M(12),M(12),M(12),M(12),INDX
      GO TO 330
260   WRITE (9,560) M(24),M(5),M(6),INDX
      GO TO 330
270   WRITE (9,570) M(13),M(13),M(13),M(13),INDX
      INDX=INDX+10
      WRITE (9,580) M(7),M(8),M70   ,INDX
      GO TO 140
280   WRITE (9,590) M(30),M(30),M(30),INDX
      GO TO 330
290   WRITE (9,600) M(29),M(29),M(29),M(29),M(29),INDX
      INDX=INDX+10
      WRITE (9,610) M(26),M(26),M(26),M(26),INDX
      INDX=INDX+10
      WRITE (9,620) M(26),M69,M61,M(10),INDX
      GO TO 220
300   WRITE (9,630) M(18),M(18),M(23),M(23),INDX
      INDX=INDX+10
      WRITE (9,640) M(23),M(20),M(20),M(20),INDX
      INDX=INDX+10
      WRITE (9,650) M(20),M(16),INDX
      GO TO 220
310   WRITE (9,660) M(22),M(22),M(22),M(22),INDX
      INDX=INDX+10
      WRITE (9,670) M(25),M(25),M(25),M(25),INDX
      INDX=INDX+10
      WRITE (9,680) M(25),M5,M(19),INDX
      GO TO 220
320   WRITE (9,360) TAPIN,INDX
330   CONTINUE
340   WRITE (6,690) J
      END FILE 9
      STOP
C
C
C
C
350   FORMAT (I4)
360   FORMAT (18A4,I8)
370   FORMAT (6X,'COMMON /COM02/ ATRIB(3),JTRIB(2),QSET(',I3,'),NSET(',I
```

## Table 3 (Continued)

```
      13,'),MFA,MXX,',5X,I8)
380   FORMAT (5X,'*',15X,'MFE(3),MLE(3),NQ(3)',32X,I8)
390   FORMAT (6X,'COMMON /COM09/ PARAM(',I3,',5),NPTBU(',I3,'),PARM1(',I
      13,'),PARM4(',I3,')',6X,I8)
400   FORMAT (6X,'COMMON /COM10/ CACIN(',I3,',3),ITCHR(',I3,'),LLTSK(',I
      13,',2),LSINK(',I3,'),',3X,I8)
410   FORMAT (5X,'*',15X,'MACIN(',I3,',4),MFEN(',I3,'),MFSTT(',I3,'),NFT
      1BU(',I3,'),',6X,I8)
420   FORMAT (5X,'*',15X,'MOP(',I3,'),MPO(',I3,'),NDCH(',I3,',2),NDEL(',
      1I3,'),NDPT(',I3,'),',1X,I8)
430   FORMAT (5X,'*',15X,'NPAR(',I3,'),NPO(',I3,'),NPODR(',I3,'),NPODS('
      1,I3,'),NPOP(',I3,'),',I8)
440   FORMAT (5X,'*',15X,'NPOR(',I3,'),NPSGN(',I3,'),NREL(',I3,'),NRELP(
      1',I3,'),',9X,I8)
450   FORMAT (5X,'*',15X,'NREL2(',I3,'),NSIGN(',I3,'),NTC(',I3,'),NTCHR(
      1',I3,'),',9X,I8)
460   FORMAT (5X,'*',15X,'NTYPE(',I3,'),KMARK(',I3,'),XMARK(',I3,')',19X
      1,I8)
470   FORMAT (6X,'COMMON /COM11/ BUSY(',I2,'),LLRES(',I2,',2),NBUS(',I2,
      1'),NOPTR(',I2,'),TLST(',I2,'),',2X,I8)
480   FORMAT (5X,'*',15X,'NOPA(',I3,'),RSTAT(',I4,')',30X,I8)
490   FORMAT (6X,'COMMON /COM12/ YABA(',I4,'),NABA(',I3,'),STCHR(',I3,')
      1',20X,I8)
500   FORMAT (6X,'COMMON /COM13/ MDFNS(',I2,'),MFSTW(',I2,'),MFSTU(',I3,
      1')',21X,I8)
510   FORMAT(6X,'COMMON /COM14/ NSINK(',I4,'),KSTPE(',I4,'),KSTTM(',I4,
      1'),XSTUS(',I4,'),',3X,I8)
520   FORMAT(5X,'*',15X,'NCELS(',I4,'),XLOW(',I4,'),WIDTH(',I4,'),SUMAI(
      1',I4,',5),',2X,I8)
530   FORMAT(5X,'*',15X,'SUMAF(',I4,',5),JCELS(',I4,')',26X,I8)
540   FORMAT (6X,'COMMON /COM15/ DESCR(',I4,'),DOATT(',I3,'),NDSTR(',I3,
      1'),SYSAT(',I3,')',7X,I8)
550   FORMAT (6X,'COMMON /COM17/ SS(',I3,'),SSL(',I3,'),DD(',I3,'),DDL('
      1,I3,'),LLSVR(',I3,',2)',5X,I8)
560   FORMAT (6X,'COMMON /COM18/ IS(',I2,'),NABAD(',I3,'),YABAR(',I3,')'
      1,23X,I8)
570   FORMAT (6X,'COMMON /COM19/ LFLAG(',I2,'),NPOSS(',I2,'),NPOST(',I2,
      1'),LLMON(',I2,',2),',9X,I8)
580   FORMAT (5X,'*',15X,'NABAT(',I4,'),NABAS(',I4,'),THRES(',I3,')',17X
      1,I8)
590   FORMAT (6X,'COMMON /COM20/ NSTAI(',I2,'),LLSVS(',I2,',2),SSTPV(',I
      12,',6)',18X,I8)
600   FORMAT (6X,'COMMON /COM21/ OTPLT(',I2,'),IITAP(',I2,'),NNPTS(',I2,
      1'),NNVAR(',I2,'),NNVP(',I2,'),',2X,I8)
610   FORMAT (5X,'*',15X,'LLPLT,NNPT,LLPHI(',I2,'),LLPLO(',I2,'),LLSYM('
      1,I2,'),PPHI(',I2,'),',1X,I8)
620   FORMAT (5X,'*',15X,'PPLO(',I2,'),NVP(',I3,'),LLSVP(',I2,',2),QPSET
      1(',I4,')',10X,I8)
630   FORMAT (6X,'COMMON /COM23/ LLUGC(',I2,',2),USOBV(',I2,',5),LLUGT('
      1,I2,',2),TTCLR(',I2,'),',5X,I8)
640   FORMAT (5X,'*',15X,'USTPV(',I2,',6),LLUGH(',I2,',2),NNCEL(',I2,'),
      1HHLOW(',I2,'),',7X,I8)
650   FORMAT (5X,'*',15X,'HHWID(',I2,'),JJCEL(',I3,')',31X,I8)
660   FORMAT (6X,'COMMON /COM24/ DPLOT(',I2,'),ITAPE(',I2,'),NPTSV(',I2,
      1'),NVARS(',I2,'),LPLOT,',5X,I8)
670   FORMAT (5X,'*',15X,'NPTEX,LPHIH(',I2,'),LPLOW(',I2,'),LSYMB(',I2,'
      1),PHIH(',I2,'),',6X,I8)
680   FORMAT (5X,'*',15X,'PLOW(',I2,'),LLUGP(',I2,',2),UPSET(',I4,')',19
      1X,I8)
```

Table 3 (Continued)

```
690    FORMAT ('J=',I5)
C
C
       END
//GO.FT08F001  DD  UNIT=TAPE,VOL=SER=SAINT,DISP=OLD,LABEL=(1,NL),
//     DCB=(RECFM=FB,BLKSIZE=80,LRECL=80)
//GO.FT09F001  DD  UNIT=DISK,VOL=SER=PUBLIC,DSN=NEWSAINT,
//     DISP=(NEW,KEEP),SPACE=(TRK,(10,5),RLSE,CONTIG),
//     DCB=(RECFM=FB,LRECL=80,BLKSIZE=6400)
//GO.SYSIN  DD  *
  20                                       ID
  20                                       IMN
  25                                       IMNA
 100                                       MAXDS
  30                                       MDAD
  60                                       MDDR
   4                                       MDNPT
   6                                       MDNSS
  10                                       MDOAT
   2                                       MDQP
 100                                       MDSTR
   5                                       MEQT
   2                                       MFLAG
   5                                       MMDFN
  50                                       MMSTU
 200                                       MNCEL
 200                                       MNCLS
  10                                       MNCLT
   2                                       MNCUP
  10                                       MNHIS
  20                                       MNOPA
  10                                       MNPLT
  10                                       MNSTP
   2                                       MNSWA
  10                                       MNVAR
  10                                       MNVPP
   2                                       MOPNO
  20                                       MPARM
  10                                       MPLOT
   2                                       MSTAT
   5                                       MSYAT
  40                                       MTCHR
  10                                       MXSTA
 150                                       MYABA
```

# REFERENCES

Duket, Steven D., David B. Wortman and Deborah J. Seifert, June 1978, Documentation for the SAINT Simulation Program, AMRL-TR-77-63 (AD A059 198), Air Force Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

Wortman, David B., Steven D. Duket, Deborah J. Seifert, Reuben L. Hann and Gerald P. Chubb, June 1978, The SAINT User's Manual, AMRL-TR-77-62 (AD A058 724), Air Force Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

Wortman, David B., Steven D. Duket, Deborah J. Seifert, Reuben L. Hann and Gerald P. Chubb, July 1978, Simulation Using SAINT: A User-Oriented Instruction Manual, AMRL-TR-77-61 (AD A058 671), Air Force Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

*U.S.Government Printing Office: 1980 — 657-084/550